**Курсовая работа по информатике  
Тема: «Определение радиуса и центра окружности  
минимального радиуса, проходящей  
хотя бы через три различные точки  
заданного множества точек»**

Содержание

1. Постановка задачи 3
2. Уточнение входных и выходных данных и ограничений на них 3
3. Метод решения задачи 3
   1. Анализ исходных данных и выбор используемой структуры данных 3
   2. Выбор метода решения 3
   3. Тесты, проверяющие правильность решения 4
4. Составление алгоритма 6
   1. Обобщённая блок-схема алгоритма 6
   2. Код программы 6

**Постановка задачи**

По условию одним из трёх способов (вручную с клавиатуры, случайным образом или из готового файла) вводятся координаты точек на плоскости. Требуется найти окружность с наименьшим радиусом, проходящую не менее, чем через три данные точки, а также найти центр и радиус этой окружности.

**Уточнение входных и выходных данных и ограничений на них**

Ограничение на координаты точек вводится самим пользователем. Точки имеют вид

Вводится число (по умолчанию равное 100), и для всех точек выполняется условие

Перед визуализацией также необходимо ввести масштаб сетки . По умолчанию . Это значит, что каждая клетка в сетке имеет размеры 10 x 10.

На выходные данные ограничений нет.

**Метод решения задачи**

**Анализ исходных данных и выбор используемой структуры данных**

Наиболее важные используемые структуры:

* Класс Vector2(x, y) — описывает точку и некоторые потребовавшиеся методы.
* Класс Circumference(Center[Vector2], Radius) —описывает окружность.
* Массив POINTS — хранит введённые точки.
* Глобальная переменная n — хранит количество введённых точек.
* Глобальная переменная width — ограничение на координаты точек.
* Глобальная переменная ratio — показывает, во сколько раз размер холста больше чем длина осей (равная width \* 2)
* Функция drawLine (ctx, x1, y1, x2, y2) — рисует на холсте отрезок с границами в точках (x1, y1) и (x2, y2).
* Функция getCenterOfCircumscribedCircle (pA, pB, pC) — находит координаты центра окружности, описанной вокруг треугольника с вершинами в точках pA, pB и pC.
* Функция task(ctx, div, canvas) — выполняет задание

**Выбор метода решения**

Циклы в функции task осуществляют перебор всех возможных сочетаний точек по три. Рассмотрим, что происходит с каждым отдельно взятым набором.

Первым математическим шагом становится нахождение центра окружности, описанной вокруг треугольника с вершинами в данных точках. Находятся длины отрезков с краями в данных точках и проверяется, не лежат ли эти точки на одной прямой. Если лежат, то последующие шаги пропускаются, и функция возвращает null. Последствия этого будут разобраны далее. Если же эти точки составляют треугольник, то далее ищутся коэффициенты прямых, содержащих его стороны, затем коэффициенты прямых, перпендикулярных сторонам и середины сторон.

Следующим шагом становится проверка коэффициентов перпендикулярных прямых на равенство бесконечности (коэффициент равен бесконечности, если прямая вертикальна). В случае вертикальности одной из прямых можно сразу заключить, что X координата центра окружности равна X координате середины отрезка, перпендикуляр к которому вертикален. Ищется уравнение серединного перпендикуляра к любой другой стороне, и уже известная x координата подставляется в найденное уравнение. Тем самым мы находим Y координату центра.

Если же ни один из перпендикуляров не вертикален, то находятся коэффициенты b в уравнениях серединных перпендикуляров с помощью известных координат середин сторон. Два серединных перпендикуляра пересекаются в центре описанной окружности, и, решая систему двух линейных уравнений, мы находим его координаты, которые возвращаются функцией в виде Vector2.

Покинув функцию, мы возвращаемся наружу в функцию task и проверяем, не равен ли возвращённый центр null. Если равен, то цикл переходит на следующий шаг. Если же нет, то мы проверяем, была ли окружность найдена раньше (т.е. не равна ли она undefined).

Если текущая окружность — первая из всех, то она помещается в память для дальнейших сравнений. Если окружность уже была обнаружена раньше, то текущая окружность сравнивается с окружностью в памяти, и выбирается та, чей радиус меньше.

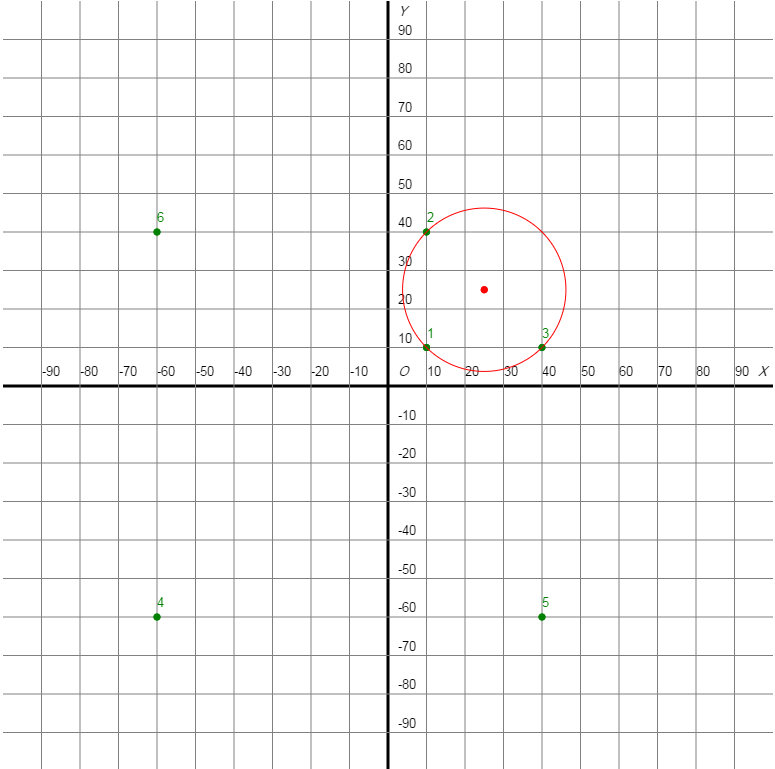
После этих операций цикл переходит на следующий шаг.

По завершению циклов в памяти остаётся окружность с наименьшим радиусом, которая и отображается на холсте.

**Тесты, проверяющие правильность решения**

Тест №1:

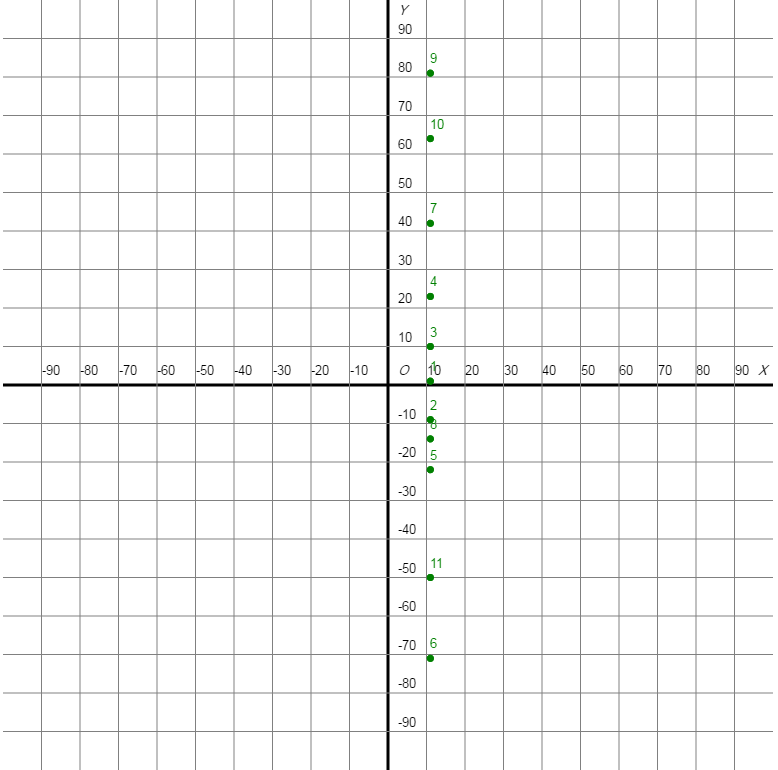
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| N | 1 | 2 | 3 | 4 | 5 | 6 |
| X | 10 | 10 | 40 | -60 | 40 | -60 |
| Y | 10 | 40 | 10 | -60 | -60 | 40 |

****

Результат теста №1: окружность с центром в точке (25; 25) и радиусом 21.21

Тест №2:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| X | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 |
| Y | 1 | -9 | 10 | 23 | -22 | -71 | 42 | -14 | 81 | 64 | -50 |

  
Результат теста №2: невозможно создать ни одной окружности через данные точки

**Составление алгоритма**

**Обобщённая блок-схема алгоритма**

**Код программы**

Выполнить задание

Ввод вручную

Случайный ввод

Визуализация

Ввод

Исправить

Вывод

Меню программы

См. файлы index.html, main.js, algebra.js, util.js.